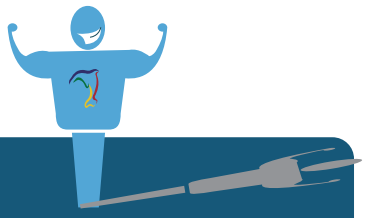
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

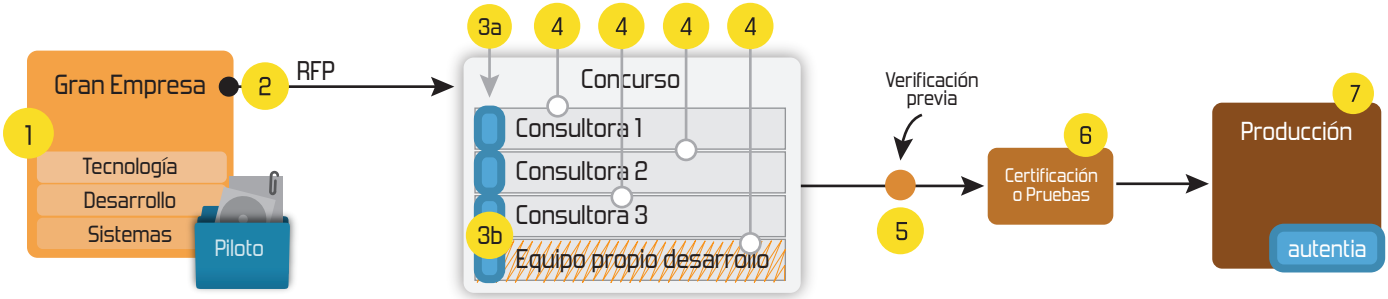
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conociemiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Tutorial desarrollado por

Carlos García Pérez

Creador del pionero Web MobileTest.

Consultor tecnológico en el desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática *

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Descargar este documento en formato PDF: xmlEncryption.pdf

Fecha de creación del tutorial: 2008-04-03

XML Encryption, Criptografía sobre XML

En este tutorial vamos a ver ejemplos de cómo realizar la encriptación y desencriptación de una sección de un documento XML, dejando el resto del documento sin encriptar.

Introducción a XML Encryption

La privacidad o confidencialidad de la información es uno de los requisitos que se deben cubrir en la gran mayoría de las aplicaciones. Debido al auge de XML como formato de representación (por ejemplo, mensajes entre servicios Web) se hace necesario mecanismos que nos permitan ocultar la información sensible.

XML Encryption **es un lenguaje** definido por W3C que nos permite especificar que partes de la información deseamos que vaya cifrada y que partes no.

El cifrado/descifrado se puede realizar tanto con claves simétricas cómo asimétricas. Recuerde que los procesos de cifrado/descrifrado básado en claves simétricas tienen muchísimo más rendimiento que los basados en claves asimétricas (pública/privada).

Para este tutorial utilizaremos la implementación de Apache (http://xml.apache.org/security/dist/).

Veamos unos ejemplos autocomentados:

1. Encriptación de parte de una información XML.
2. Desencriptación de la información anterior.

Documento XML para los ejemplos:

Dada la siguiente información en XML, vamos a encriptar exclusivamente mediante criptografía simétrica la información relaccionada con la **tarjeta de crédito**.

Últimos tutoriales

2008-04-05
JMX y monitorización de JBoss

2008-04-05
Jersey: la implemetación de RESTFull de Sun

2008-04-05
Metro: pila de webservices de Sun. Integración con Maven 2

2008-04-05
Metro: pila de webservices de Sun.

2008-04-04
Espectaculares efectos visuales en el escritorio de Linux, con Compiz Fusion

2008-04-04
Monitorización de Web Services con Glassfish Wsmonitor

2008-04-04
Axis2. Ejemplo de creación de un servicio Web

2008-04-03
Servicios Web RESTful en Axis 2

2008-04-03
XML Signature - Firma Digital sobre XML

2008-04-03
XML Encryption, Criptografía sobre XML

view plain print ?

```
01. <persona id="468300000">
02.         <nombre>Marvis</nombre>
03.         <apellidos>Rondon Marcelo</apellidos>
04.         <email>marvis@servidor.com</email>
05.         <tarjetaCredito>
06.             <numero>83838383</numero>
07.             <fechaExpiracion>01/05</fechaExpiracion>
08.         </tarjetaCredito>
09.  </persona>
```

Clase de utilidades usada en los ejemplos de este tutorial

view plain print ?

```
package com.autentia.examples.xmlencryption;

import java.io.File;
import java.io.FileOutputStream;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;
import org.w3c.dom.Element;

/**
 * Clase de utilidad.
 * @author Carlos García Pérez. Autentia.
 * @see     http://www.mobiletest.es
 */
public class DOMUtils {

    /**
      * Serializa un objeto Document en un archivo
      */
    public static void outputDocToFile(Document doc, File file)  throws Exception {
        FileOutputStream    f              =  new FileOutputStream(file);
        TransformerFactory factory         = TransformerFactory.newInstance();
        Transformer        transformer     = factory.newTransformer();

        transformer.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION,  "yes");

        transformer.transform( new DOMSource(doc),  new StreamResult(f));

        f.close();
    }

    /**
      * Lee un Document desde un archivo
      */
    public static Document loadDocumentFromFile(java.io.File file)  throws Exception {
        DocumentBuilderFactory  factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder         builder =  null;

        factory.setNamespaceAware( true);

        builder = factory.newDocumentBuilder();

        return builder.parse(file);
    }

    /**
      * @return Crea un elemento <tag>value</tag>
      */
    public static Element createNode(Document document, String tag, String value){
        Element node = document.createElement(tag);
        if (value != null){
            node.appendChild(document.createTextNode(value));
        }
        return node;
    }
}
```

Ejemplo de encriptación de una sección de un documento XML

view plain print ?

```java
package com.autentia.examples.xmlencryption;


import java.io.File;
import java.io.FileOutputStream;
import java.security.Key;
import javax.xml.parsers.*;
import javax.crypto.SecretKey;
import javax.crypto.KeyGenerator;
import org.apache.xml.security.keys.KeyInfo;
import org.apache.xml.security.encryption.XMLCipher;
import org.apache.xml.security.encryption.EncryptedData;
import org.apache.xml.security.encryption.EncryptedKey;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

/**
 * Este ejemplo encripta la información relacionada con la tarjeta de crédito de un cliente
 * @author Carlos García Pérez. Autentia.
 * @see     http://www.mobiletest.es
 */
public class Encrypter {
    private static final String SECRET_KEY_FILENAME =  "mykey.dat";
    private static final String ENCRYPTED_XML_FILENAME    =  "infoCifrada.xml";

    /**
      * Genera un Document de ejemplo
      */
    private static Document createSampleDom()  throws Exception {
        DocumentBuilderFactory  factory  = DocumentBuilderFactory.newInstance();
        DocumentBuilder         builder  = factory.newDocumentBuilder();
        Document                document = builder.newDocument();

        Element person = document.createElement( "persona");
        person.setAttribute( "id", "468300000");

        person.appendChild(DOMUtils.createNode(document,  "nombre",   "Marvis"));
        person.appendChild(DOMUtils.createNode(document,  "apellidos", "Rondon Marcelo"));
        person.appendChild(DOMUtils.createNode(document,  "email",       "marvis@servidor.com"));

        Element creditCard = document.createElement( "tarjetaCredito");
        creditCard.appendChild(DOMUtils.createNode(document,  "numero",   "83838383"));
        creditCard.appendChild(DOMUtils.createNode(document,  "fechaExpiracion",  "01/05"));

        person.appendChild(creditCard);

        document.appendChild(person);


        return document;
    }


    /**
      * @return Genera la clave secreta que servirá para encriptar/desencriptar la información
      * @throws Exception Cualquier incidencia
      */
    private static SecretKey generateAndStoreKeyEncryptionKey()  throws Exception {
        // Generamos la clave usando el algoritmo Triple DES
        KeyGenerator keyGenerator = KeyGenerator.getInstance( "DESede"); // Algoritmo JCE: Triple DES
        SecretKey    secret       = keyGenerator.generateKey();
        byte[]       bytes        = secret.getEncoded();

        // Guardamos la clave en disco
        File            keyFile =  new File(SECRET_KEY_FILENAME);
        FileOutputStream output  =  new FileOutputStream(keyFile);
        output.write(bytes);
        output.close();

        System.out.println( "La clave de encriptación está guardada en: "  + keyFile.getAbsolutePath());

        return secret;
    }

    /**
      * @return Devuelve la clave de encriptación de datos
      * @throws Exception Cualquier incidencia
      */
    private static SecretKey generateDataEncryptionKey()  throws Exception {
        KeyGenerator keyGenerator = KeyGenerator.getInstance( "AES");    // Algoritmo JCE: Advanced Encryptic
        keyGenerator.init( 128);
        return keyGenerator.generateKey();
    }


    /**
      * Punto de entrada del ejemplo
      * @throws Exception Cualquier incidencia
      */
    public static void main(String args[]) throws Exception {
        // Inicializamos el FrameWork de seguridad de Apache a los valores por defecto
        org.apache.xml.security.Init.init();

        Document document = Encrypter.createSampleDom();

        // Obtenemos la clave para encriptar el elemento.
        Key symmetricKey = Encrypter.generateDataEncryptionKey();
```

Documento encriptado (`infoCifrada.xml`)

En el siguiente documento se muestra la información que sería enviada al receptor. En ella se puede observar entre otras cosas:

1. La información relacionada con la tarjeta de crédito ha sido sustituida por el elemento **`xenc:EncryptedData`**
2. El elemento **`xenc:EncryptionMethod`** contiene información sobre el método de encriptación usado.
3. El elemento **`KeyInfo`** contiene información sobre la clave de encriptación.
4. El elemento **`CipherData`** contiene la información encriptada, es decir, la información sobre la tarjeta de crédito.
5. El elemento **`EncryptedKey`** contiene la clave de encriptación/desencriptación de la información almacenada en **`CipherCipherValue`**.

view plain print ?

```
<persona id="468300000">
    <nombre>Marvis</nombre>
    <apellidos>Rondon Marcelo</apellidos>
    <email>marvis@servidor.com</email>
    <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        Type="http://www.w3.org/2001/04/xmlenc#Element">
        <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"
            xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" />
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <xenc:EncryptedKey
                xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
                <xenc:EncryptionMethod
                    Algorithm="http://www.w3.org/2001/04/xmlenc#kw-tripledes"
                    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" />
                <xenc:CipherData
                    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
                    <xenc:CipherValue
                        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
                        gMp/3ZuYVyHn74JDKr3WCLDrf7H+S6wLqGEdRdgqQGw=
                    </xenc:CipherValue>
                </xenc:CipherData>
            </xenc:EncryptedKey>
        </ds:KeyInfo>
        <xenc:CipherData
            xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
            <xenc:CipherValue
                xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
                Wr1njyJlYYOM9lAYqcwGCWkw2L4pUjQD2GGVoU9lVZ0wKqHY8y3l3GY8FY4i5K3G8grIe2xN4u7x
                7RtkFiXZgGMeYnQp6oB6ckKp3KFKHVqtucc9AVzOgC7XAw/oe61HRFqe6RRVzXjNMLU5TaV7lJF1
                I8NVPQmUSDX7NRtnR68=
            </xenc:CipherValue>
        </xenc:CipherData>
    </xenc:EncryptedData>
</persona>
```

Ejemplo de desencriptación del archivo (`infoCifrada.xml`)

view plain print ?

```java
package com.autentia.examples.xmlencryption;

import java.io.File;

import java.security.Key;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.DESedeKeySpec;
import org.apache.xml.security.encryption.XMLCipher;
import org.apache.xml.security.utils.JavaUtils;
import org.apache.xml.security.utils.EncryptionConstants;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

/**
 * Este ejemplo desencripta la información encriptada de un documento XML
 * @author Carlos García Pérez. Autentia.
 * @see    http://www.mobiletest.es
 */
public class Decrypter {
    private static final String SECRET_KEY_FILENAME       =  "mykey.dat";
    private static final String ENCRYPTED_XML_FILENAME    =  "infoCifrada.xml";
    private static final String DESENCRYPTED_XML_FILENAME =  "infoDescifrada.xml";


    /**
      * @return La clave de encriptación/encriptación desde un archivo
      * @throws Exception Cualquier incidencia
      */
    private static SecretKey loadDesencryptionKey()  throws Exception {
        DESedeKeySpec    keySpec =  new DESedeKeySpec(JavaUtils.getBytesFromFile(SECRET_KEY_FILENAME));
        SecretKeyFactory skf     = SecretKeyFactory.getInstance( "DESede");
        SecretKey        key     = skf.generateSecret(keySpec);

        return key;
    }

    /**
      * Punto de entrada del ejemplo
      * @throws Exception Cualquier incidencia
      */
    public static void main(String args[]) throws Exception {
         // Inicializamos el FrameWork de seguridad de Apache a los valores por defecto
        org.apache.xml.security.Init.init();

        // Obtenemos el documento xml encriptado
        Document document = DOMUtils.loadDocumentFromFile( new File(ENCRYPTED_XML_FILENAME));

        // Accedemos al nodo con la información encriptada.      namespace: "http://www.w3.org/200
        Element node = (Element) document.getElementsByTagNameNS(EncryptionConstants.EncryptionSpecNS, EncryptionCo
        Key     kek  = Decrypter.loadDesencryptionKey();     // Carga la clave para desencriptar la informa

        // La clave que será usada para desencriptar los datos del xml se obtendrá desde el KeyInf
        XMLCipher cipher = XMLCipher.getInstance();
        cipher.init(XMLCipher.DECRYPT_MODE,  null);  // Key=null para que use como clave el EncryptedKey
        cipher.setKEK(kek);

        // Desencriptamos reemplazando los datos encriptados con su contenido desencriptado
        cipher.doFinal(document, node);

        // Guarda el Document en un archivo
        File file = new File(DESENCRYPTED_XML_FILENAME);
        DOMUtils.outputDocToFile(document, file);

        System.out.println( "Los datos han sido desencriptados en: "  + file.toURL().toString());
    }
}
```

Saludos, Carlos García.

- Puedes opinar sobre este tutorial haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati. ADD THIS BLOG TO MY Technorati FAVORITES

Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subcripción a novedades:

E-mail [ ] Aceptar

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Procesamiento XML en Java con JAXB y WSDP 1.6	Os mostramos como instalar la versión 1.6 de WSDP y como procesar los ficheros XML con uno de sus componentes, JAXB	2005-07-09	11562	pdf
XMLBeans, una forma de mapear un XML en objetos Java	En este tutorial vamos a ver una introducción a XMLBeans para ver como podemos obtener, a partir de un DTD o un XSD, las clases Java que procesan los XML que cumplen ese DTD o ese XSD.	2007-08-20	3114	pdf
Soporte XML en Eclipse con X-MEN	Alejandro Perez nos enseña como potenciar el entorno eclipse para facilitarnos el trabajo con ficheros xml, gracias al pluggin X-MEN	2003-12-27	17077	pdf
XML básico	Si quieres ver de un modo visual como crear un documento XML, este es tu tutorial. Este es el primero de un conjunto de tutoriales que iremos publicando sobre esta fascinante y amplia tecnología	2003-06-10	21700	pdf
Schemas XML. Introducción esquemas XML	Los esquemas XML (schemas XML) son una evolución natural de las DTDs. Os mostramos como emprezar con esta tecnología.	2003-12-17	17625	pdf
XML y XSL en Cliente	En este tutorial os enseñamos como formaterar documentos XML directamente en vuestro navegador a través de Plantillas XSL. En cursos sucesivos veremos como hacerlo en el servidor, para no crear dependencias con el navegador del cliente.	2003-06-11	16069	pdf
Usar DataSource XML para crear informes con iReport	Este tutorial nos enseña como poder crear informes usando un datasource a partir de un fichero XML. También nos dirá como poder crear subinformes con este mismo tipo de conexion/fuente de datos	2007-10-26	2646	pdf
Transformación de XML y XSL en JSPs	Os mostramos como poder utilizar XML y XSL en JSPS, combinado con el Patrón MVC	2003-12-06	25683	pdf
XMLEncryption en Java	En este magnífico tutorial, Alberto Carrasco nos enseña los fundamentos y un ejemplo práctico de XMLEncryption.	2005-11-24	8440	pdf